

Müller HM Data Framescanner



Reference manual
DSP

Version history

Version	Date	Changes	Author	Distribution
1.0	02-02-2011	Initial version	Roel Eckhardt	Customers Müller HM dataframescanner

Index

1.	Quick guide	5
2.	Technical requirements PC	6
3.	Software installation	7
3.1	VirtualDub	7
3.2	AviSynth	7
3.3	Müller HM DSP	7
4.	How to use the Müller HM DSP	8
4.1	Edit the DSP script	8
4.2	Drag the DSP file into VirtualDub	11
4.3	Save new AVI to output location	11

About Müller HM DSP

FilmFabriek supplies Open Source software used with AviSynth executed by VirtualDub. This Open Source software can be used for everyone in post-processing activities like restoration & enhancement such as de-noising, stabilization, coloring, sharpening, de-graining, etc.

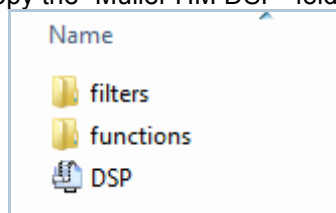
The basis for the AviSynth script is made by all the open source developers who have made fantastic filters for everyone to use like Theodor Anschütz, A.G. Balakhnin aka Fizick, Reiner Wittman and all others. A big credit goes out to them and their work.

The Müller HM DSP is a mixture of combining these filters, the individual filter settings and parameters and finally finding the right processing order.

In order to use this software you must install AviSynth and VirtualDub.

1. Quick guide

1. install VirtualDub
2. install AviSynth
3. Copy the "Muller HM DSP" folder to any location. The contents should look like this:



4. Edit the DSP file with notepad and edit the desired AVI file and location or directory where the bitmap files are saved. Then Save file

```
#  
# 1. Choose source - BMP or AVI  
#  
#source = read_avi("E:\AVI\reel1.avi")  
source = read_bmp(18,0,006628,"E:\BMP\reel1") # fps, firstframe, lastframe, directory
```

5. Start VirtualDub
6. Drag the DSP file into VirtualDub
7. select file > save as AVI (or F7)
8. select output location for .AVI file.

2. Technical requirements PC

The Müller HM DSP uses a variety of standard open source filters like de-noising, stabilization, coloring, sharpening etc. This process requires a lot of CPU power. A powerful CPU is required to minimize restoration-processing time. Of course processing can be done overnight. Using less stabilization, less de-noising, less sharpening etc. reduces the processing time.

A slow CPU will not affect the outcome but only the processing time.

3. Software installation

3.1 VirtualDub

VirtualDub is a video capture and video processing utility for Microsoft Windows written by Avery Lee. It is designed to process linear video streams, including filtering and recompression. Released under the terms of the GNU General Public License, VirtualDub is free software. VirtualDub operates on AVI files or picture files like .BMP's, requiring that appropriate video and audio codecs are installed.

Install the VirtualDub Software from <http://www.virtualdub.org>

3.2 AviSynth

AviSynth is a powerful tool for video post-production. It provides ways of editing and processing videos. AviSynth works as a frameserver, providing instant editing without the need for temporary files.

AviSynth itself does not provide a graphical user interface (GUI), but instead relies on a script system that allows advanced non-linear editing. While this may at first seem tedious and unintuitive, it is remarkably powerful and is a very good way to manage projects in a precise, consistent, and reproducible manner. Because text-based scripts are human readable, projects are inherently self-documenting. The scripting language is simple yet powerful, and complex filters can be created from basic operations to develop a sophisticated palette of useful and unique effects.




Install the AviSynth Software from <http://avisynth.org>

3.3 Müller HM DSP

The Müller HM DSP contains a number of filters and avisynth scrips (.avs)

Müller HM DSP doesn't require a typical windows installation, you just copy it to any desired location.


After copying, it would look something like this:




Name	Date modified	Type	Size
 filters	1/2/2011 9:18 PM	File folder	
 functions	1/2/2011 9:35 PM	File folder	
 DSP	1/2/2011 9:41 PM	AviSynth Script	9 KB

4. How to use the Müller HM DSP

4.1 Edit the DSP script

Edit the file “DSP” with a text editor like notepad.



Name	Date modified	Type	Size
 filters	1/2/2011 9:18 PM	File folder	
 functions	1/2/2011 9:35 PM	File folder	
 DSP	1/2/2011 9:41 PM	AviSynth Script	9 KB

Section 1: enter the AVI filename or BMP directory location

You can either read an AVI file or read the individual scanned .BMP files directly into the DSP

1. if you use an AVI file as source then use the read_avi function and enter the location of the avi file
2. if you prefer to read the individual .BMP files directly then use the read_bmp function. The parameters are: framerate, start frame, end frame, source directory.

**The '#' is special character which ignores everything behind the '#'.
#**

Remember to save your settings

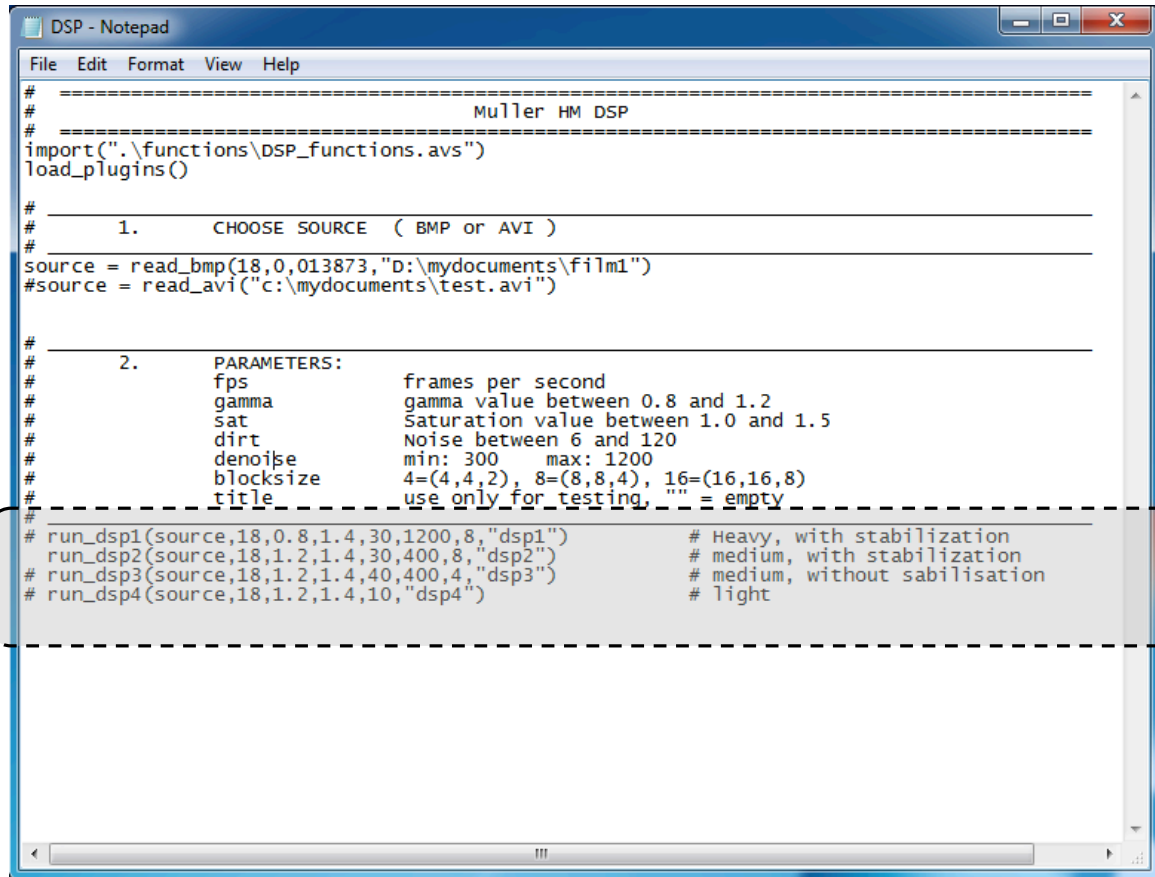
```
# DSP - Notepad
# File Edit Format View Help
# ===== Muller HM DSP =====
#
import(".\functions\DSP_functions.avs")
load_plugins()

#
# 1. CHOOSE SOURCE ( BMP or AVI )
#
source = read_bmp(18,0,013873,"D:\mydocuments\film1")
#source = read_avi("c:\mydocuments\test.avi")

#
# 2. PARAMETERS:
# fps frames per second
# gamma gamma value between 0.8 and 1.2
# sat Saturation value between 1.0 and 1.5
# dirt Noise between 6 and 120
# denoise min: 300 max: 1200
# blocksize 4=(4,4,2), 8=(8,8,4), 16=(16,16,8)
# title use only for testing, "" = empty
#
# run_dsp1(source,18,0.8,1.4,30,1200,8,"dsp1") # Heavy, with stabilization
# run_dsp2(source,18,1.2,1.4,30,400,8,"dsp2") # medium, with stabilization
# run_dsp3(source,18,1.2,1.4,40,400,4,"dsp3") # medium, without sabilisation
# run_dsp4(source,18,1.2,1.4,10,"dsp4") # light
```

Section2: choose desired output

There are four versions: DSP1,DSP2, DSP3 and DSP4. Choose the one that best fits your own personal choice. Remember to save your settings. DSP2 gives a good result with stabilization. Be sure to try out the different versions.



```
# =====
#                               Muller HM DSP
# =====
import("..\functions\DSP_functions.avs")
load_plugins()

# -----
# 1.      CHOOSE SOURCE  ( BMP or AVI )
# -----
source = read_bmp(18,0,013873,"D:\mydocuments\film1")
#source = read_avi("c:\mydocuments\test.avi")

# -----
# 2.      PARAMETERS:
#          fps          frames per second
#          gamma        gamma value between 0.8 and 1.2
#          sat          Saturation value between 1.0 and 1.5
#          dirt         Noise between 6 and 120
#          denoise      min: 300    max: 1200
#          blocksize    4=(4,4,2), 8=(8,8,4), 16=(16,16,8)
#          title        use only for testing, "" = empty
# -----
# run_dsp1(source,18,0.8,1.4,30,1200,8,"dsp1")      # Heavy, with stabilization
# run_dsp2(source,18,1.2,1.4,30,400,8,"dsp2")      # medium, with stabilization
# run_dsp3(source,18,1.2,1.4,40,400,4,"dsp3")      # medium, without stabilization
# run_dsp4(source,18,1.2,1.4,10,"dsp4")            # light
```

4.2 Drag the DSP file into VirtualDub

1. Open / run VirtualDub
2. Drag the DSP script into VirtualDub.

4.3 Save new AVI to output location

1. In VirtualDub: select file > save as AVI (or F7)
2. select output location for new .AVI file output
3. press ENTER, VirtualDub is now running:

